

What is Claimed is:

1. Data processing apparatus having processing means, memory means and display means, wherein

 said processing means performs a process in response to program instructions read from said memory means via dynamically linked operational objects called by control objects, such that events are returned back to a calling control object;

5 a plurality of pages are defined in a mark-up language that are selectively displayed and executed by a controlled browser;

 said controlled browser is controlled by a controlling container object;

 active control objects for calling said operational objects are contained within

10 said container object;

 a single passthrough object is created;

 at least one of said pages includes a page embedded control object configured to call said passthrough object;

 an initiating one of said page embedded objects calls said passthrough object

15 and passes to said passthrough object output information detailing a desired call to a specified operational object;

 said passthrough object interprets output information received from a page embedded object to generate a call to a contained object that in turn calls the desired operational object; and

20 said passthrough object receives event data from a called operational object and returns input data to said initiating embedded object indicative of said returned event.

2. Apparatus according to claim 1, configured as a self service terminal.

3. Apparatus according to claim 2, wherein said self service terminal dispenses money and facilitates financial transactions.

4. Apparatus according to claim 1, wherein said mark-up language is hypertext mark-up language (HTML).

5. Apparatus according to claim 1, wherein said passthrough object is a separate process executed under the control of an operating system.

6. Apparatus according to claim 5, wherein any process configured to create a passthrough object will firstly ensure that such an object is not in existence and only create the object if it is not in existence thereby ensuring that only one passthrough object exists at any one time.

7. Apparatus according to claim 1, wherein said call to a contained object is made via a specific decoding object within said container object.

8. Apparatus according to claim 1, wherein return events are returned to an initiating page via said passthrough object.

9. Apparatus according to claim 8, wherein said passthrough object maintains a register of established pages.

10. Apparatus according to claim 9, wherein said passthrough object includes a buffer for buffering events to be returned to pages that are no longer established within the process.

11. Apparatus according to claim 10, wherein buffered events are returned to non-established pages after said pages have been re-established.

12. A method of defining a process in a computer system via a plurality of pages defined in a mark-up language that are executable by a browser, in which said process implements operations via dynamically linked operational objects called by control objects such that events are returned back to a calling control object;

5 a plurality of pages defined in a mark-up language are selectively displayed and executed by a controlled browser;

said controlled browser is controlled by a controlling container object; active control objects for controlling said operational objects are contained within said container object;

10 a single passthrough object is created; and

pages include a page embedded control object configured to call said passthrough object, said method comprising the steps of:

calling said passthrough object from an initiating embedded object and passing output information detailing a desired call to a specified operational object;

15 interpreting said output information at said passthrough object to effect a call to said operational object; and

returning input data to an initiating embedded object indicative of a returned event via said passthrough object.

13. A method according to claim 12, wherein said mark-up language is hypertext mark-up language (HTML).

14. A method according to claim 12, wherein said passthrough object is a separate process executed under the control of an operating system.

15. A method according to claim 14, wherein any process configured to create a passthrough object will firstly ensure that such an object is not in existence and only create the object if it is not in existence thereby ensuring that only one passthrough object exists at any one time.

16. A method according to claim 12, wherein said call to a contained object is made via a specific decoding object within the container.

17. A method according to claim 12, wherein return events are returned to an initiating page via said passthrough object.

18. A method according to claim 17, wherein said passthrough object maintains a register of established pages.

19. A method according to claim 18, wherein said passthrough includes a buffer for buffering events to be returned to pages that are no longer established within the process.

20. A method according to claim 19, wherein buffered events are returned to non-established pages after said pages have been re-established.

21. A computer-readable medium having computer-readable instructions executable by a computer such that, when executing said instructions, said computer will perform the steps of:

5 establishing a library of dynamically linkable objects that may be called by control objects such that events are returned back to a calling control object;

24. A computer-readable medium having computer-readable instructions according to claim 21, such that when executing said instructions, a computer will return events to an initiating page via said passthrough object.

25. A computer-readable medium having computer-readable instructions according to claim 24, such that when executing said instructions, said computer will ensure that said passthrough object maintains a register of established pages.

26. A computer-readable medium having computer-readable instructions according to claim 25, such that when executing said instructions, a computer will ensure that said passthrough object includes a buffer for buffering events to be returned to pages that are no longer established within the process.

27. A computer-readable medium having computer-readable instructions according to claim 26, such that when executing said instructions, a computer will ensure that buffered events are returned to non-established pages after said pages have been re-established.

2025 RELEASE UNDER E.O. 14176

establishing the availability of a plurality of pages defined in a mark-up language that may be selectively displayed and executed by a controlled browser, wherein said controlled browser is controlled by a controlling container object;

containing active control objects for controlling operational objects within

5 said container object; and

facilitating the establishment of a single passthrough object, wherein pages defined in said mark-up language include a page embedded control object configured to call said passthrough object such that, during a session with a user, said program computer will further perform the steps of:

10 calling said passthrough object from an initiating embedded object and passing output information detailing a desired call to a specified operational object;

interpreting said output information at said passthrough object to effect a call to said operational object; and

returning input information to an initiating embedded object indicative of a

15 returned event via said passthrough object.

22. A computer-readable medium having computer-readable instructions according to claim 21, such that when executing said instructions, said computer will execute a process configured to create a passthrough object, wherein said process will firstly ensure that such an object is not in existence and only create the object if it is not in existence

5 thereby ensuring that only one passthrough object exists at any one time.

23. A computer-readable medium having computer-readable instructions according to claim 21, such that when executing said instructions, said computer will make a call to a contained object, wherein this call is made via a specific decoding object within the container.